Step 5 Requirements

Final Presentations (10pts)

Each team will deliver a **10-minute presentation**, followed by **5 minutes for questions**. Please **rehearse** the presentation and structure it as a story. It should follow a clear narrative or storybook flow that guides the audience smoothly from beginning to end.

Your presentation must cover at least the following elements:

- A brief description of the application purpose and objectives and the challenges that have been faced.
- A brief explanation of the **development workflow**
  (from idea → UX/UI design → implementation → testing → deployment)
- An overview of the **technology stack** you selected and any specific libraries used
- A discussion of how you have implemented **best practices**, including:
  - Modular architecture
  - Reusability and separation of concerns
  - Data handling (APIs, local storage, database usage)
  - Performance and accessibility considerations
- Conclusion with a takeaway message (e.g. positive realization, constructive ideas).

All team members must actively present.
Each team member should present a specific section that they have been working on of the presentation.

Physical presence is required for all presenters.
If a team member has a valid reason for participating online, you must request approval **in advance** from the course responsible.

Time Slots for the presentation:

- Wednesday 17th Room 1021, at 12:00
- Wednesday 17th December Room 1021, at 14:00
- Friday 19th December Room 1019, at 10:00

Please also don't forget to register in OIS2 system for the Exam/Final Presentation, this applies to all of you (team leaders and team members).

Documentation (10pts)

1. GitHub Repository Requirements

Your GitHub repository must include a **Project README** containing:

- Project overview, goals, and main features
- List of team members and their GitHub handles
- Installation steps, dependencies, and build instructions
- Usage guide with screenshots or a demo GIF
- Explanation of the project structure

2. Developer Documentation (to be submitted on the course webpage)

The Developer Documentation must include:

- API specifications or data model descriptions
- Overview of key modules, services, and workflows
- User-facing instructions (if relevant), including step-by-step usage guidance

GitHub Code Quality Requirements (10pts)

Please ensure your GitHub repository is **up-to-date** and the codebase is **clean and well-structured**.

Code documentation must follow these guidelines:

- Use **KDoc** (`/** ... */`) to document:
    - Functions and parameters
    - Return values
    - Exceptions
    - (Optional) Usage examples
- Use **inline comments** only when necessary to clarify complex logic
- Follow consistent **Kotlin coding conventions**, including:
    - Kotlin style guide (naming, formatting, line length)
    - Use of linting tools such as **ktlint** or **detekt**